Azure Machine Learning - (MLAA)

Mutaz Abu Ghazaleh | AI/ML Specialist
19/04/2022

UTS   Microsoft

Agenda

- Get Azure and Azure Portal

- Machine Learning Workspace

- Development Environment
  - Options
  - Compute Instances
  - RStudio Online and Notebooks

- Code Example

- Experiments and Tracking

Get Azure

Azure Free Account

Azure for Students

https://azure.microsoft.com/en-au/free/
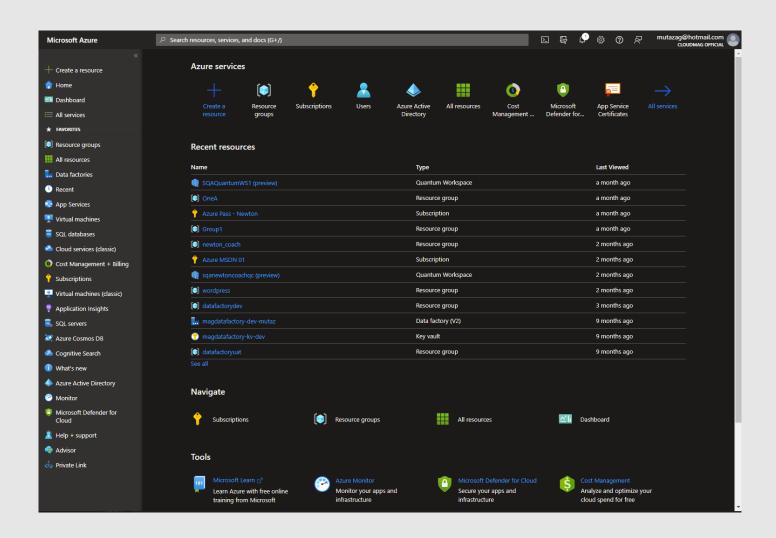
https://azure.microsoft.com/en-us/free/students/

Azure Portal

- A single portal to access all applications in your Azure Subscription
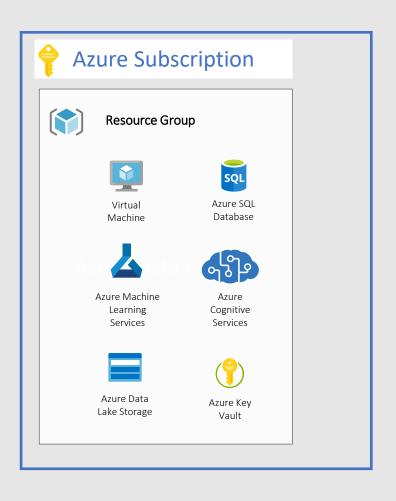
- Build, manage and monitor Azure resources

- Login with an Azure account: https://portal.azure.com

Hierarchy in Azure

Azure account: The email address that you provide when you create an Azure subscription is the Azure account for the subscription.

Subscription: A logical container for your resources. Each Azure resource is associated with only one subscription. Creating a subscription is the first step in adopting Azure.

Resource groups: Logical containers that you use to group related resources in a subscription. They're commonly used to represent a collection of assets that are required to support a workload, application, or specific function within a subscription.

Resources: An entity that's managed by Azure. Examples include Azure Virtual Machines, Machine Learning Services, SQL Database and storage accounts.

Azure Products

AI + Machine Learning

Category
AI + Machine Learning
Analytics
Compute
Containers
Databases
Developer Tools
DevOps
Hybrid + multicloud
Identity
Integration
Internet of Things
Management and Governance
Media
Migration
Mixed Reality
Mobile
Networking
Security
Storage
Virtual desktop infrastructure
Web

AI + Machine Learning

Anomaly Detector
Easily add anomaly detection capabilities to your apps

Azure Applied AI Services
Specialized services that enable organizations to accelerate time to value in applying AI to solve common scenarios

Azure Bot Service
Intelligent, serverless bot service that scales on demand

Azure Cognitive Search
AI-powered cloud search service for mobile and web app development

Azure Cognitive Services
Add smart API capabilities to enable contextual interactions

Azure Databricks
Fast, easy, and collaborative Apache Spark-based analytics platform

Azure Machine Learning
Bring AI to everyone with an end-to-end, scalable, trusted platform with experimentation and model management

Azure Open Datasets
Cloud platform to host and share curated open datasets to accelerate development of machine learning models

Azure Video Analyzer (preview)
Build intelligent video-based applications using the AI of your choice

Azure Video Analyzer for Media
Unlock video insights

Computer Vision
Distill actionable information from images

Content Moderator
Automated image, text, and video moderation

Custom Vision
Easily customize your own state-of-the-art computer vision models for your unique use case

Data Science Virtual Machines
Rich pre-configured environment for AI development

Face
Detect, identify, analyze, organize, and tag faces in photos

Form Recognizer
The AI-powered document extraction service that understands your forms

Health Bot
A managed service purpose-built for development of virtual healthcare assistants.

Immersive Reader
Empower users of all ages and abilities to read and comprehend text

Kinect DK
Build computer vision and speech models using a developer kit with advanced AI sensors

Language Understanding
Teach your apps to understand commands from your users

Metrics Advisor
An AI service that monitors metrics and diagnoses issues

Microsoft Genomics
Power genome sequencing & research insights

Personalizer
An AI service that delivers a personalized user experience
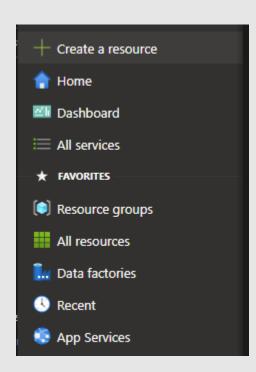
Project Bonsai (Preview)
A machine teaching service for creating intelligent industrial control systems using simulations

QnA Maker
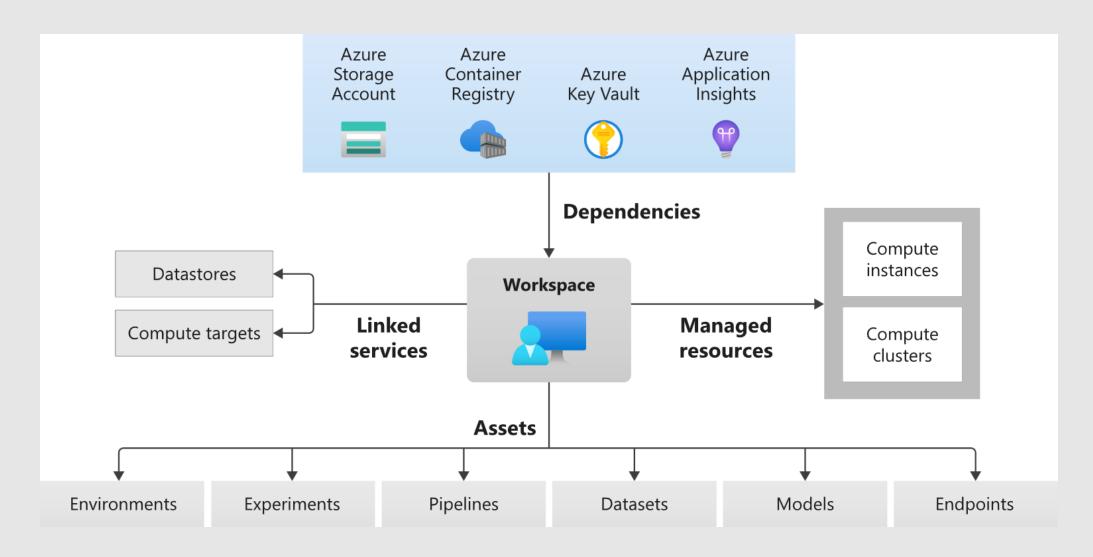Distill information into conversational, easy-to-navigate answers

Speaker Recognition (Preview)
Use speech to identify and verify individual speakers

Speech to Text
Convert spoken audio to text for more natural interactions

Speech Translation
Easily integrate real-time speech translation to your app

Text Analytics
Easily evaluate sentiment and topics to understand what users want
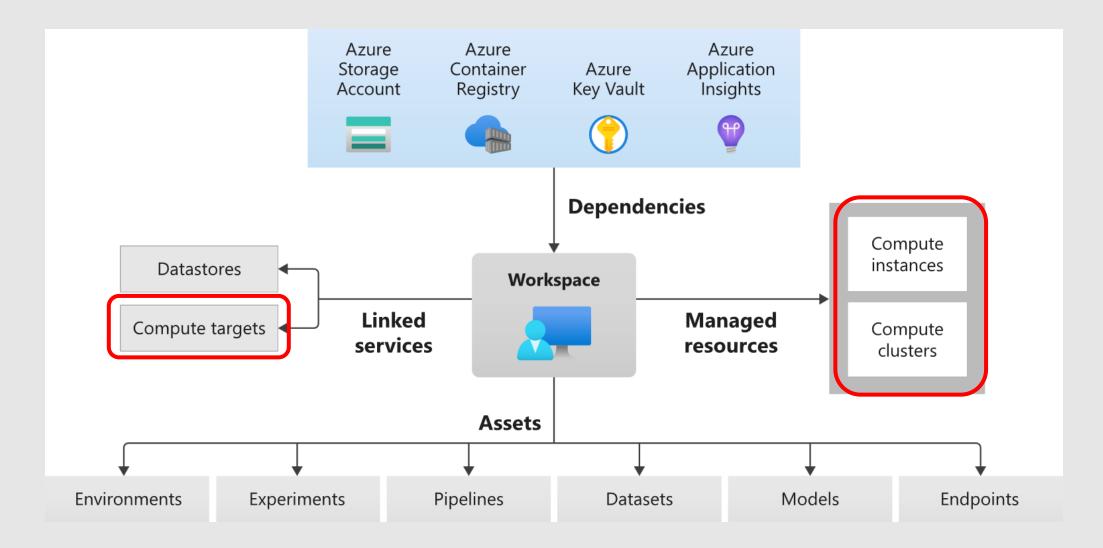
Text to Speech
Convert text to speech to create more natural, accessible interfaces

Translator
Easily conduct machine translation with a simple REST API call

(Right sidebar)

+ Create a resource

🏠 Home

📊 Dashboard

☰ All services

★ FAVORITES

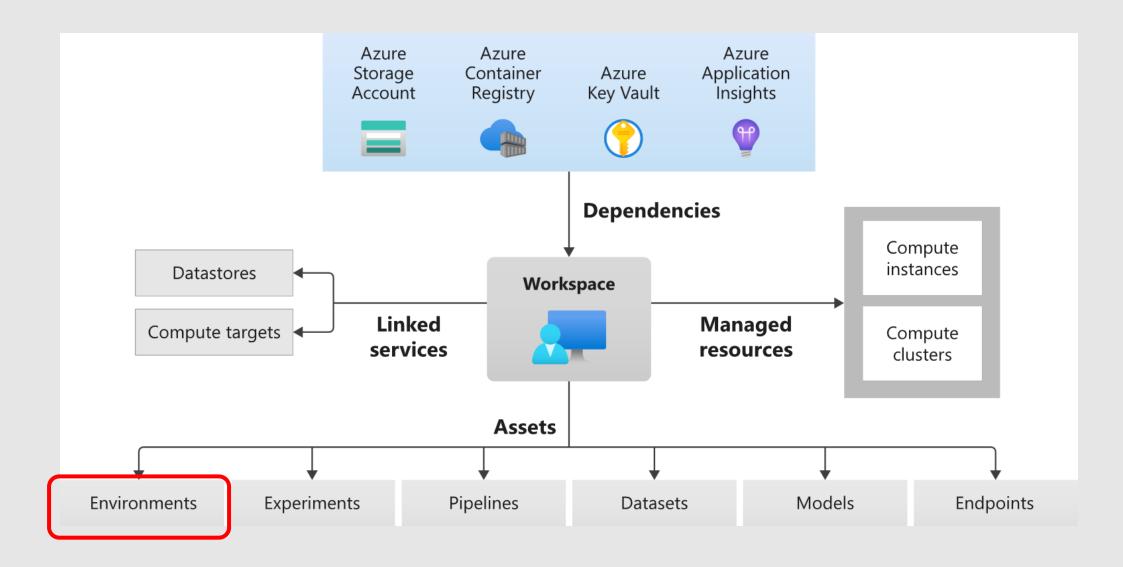Resource groups
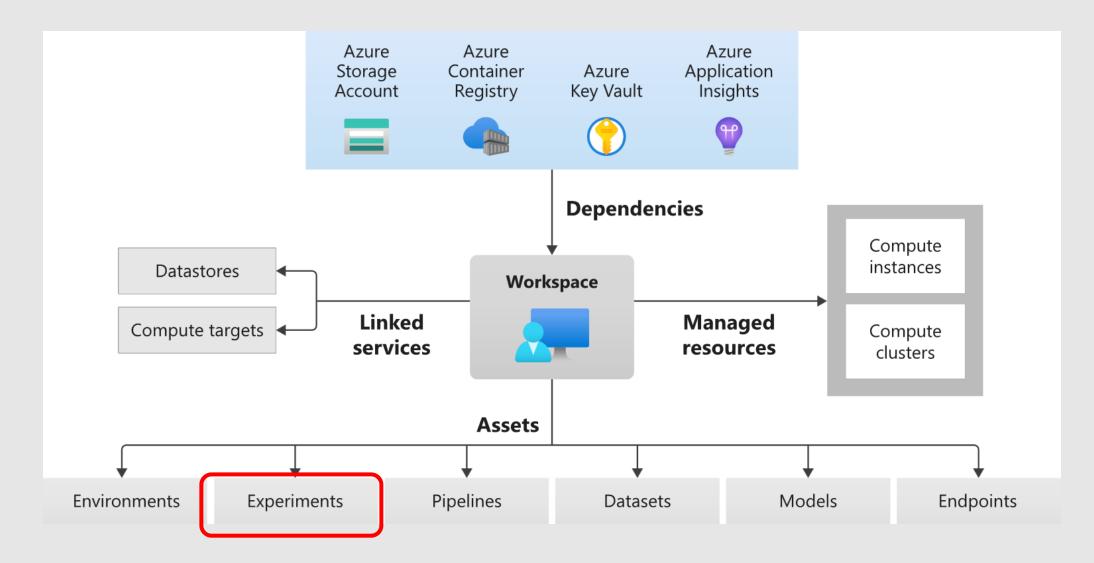
All resources

Data factories

Recent

App Services

Machine Learning Workspace

Compute
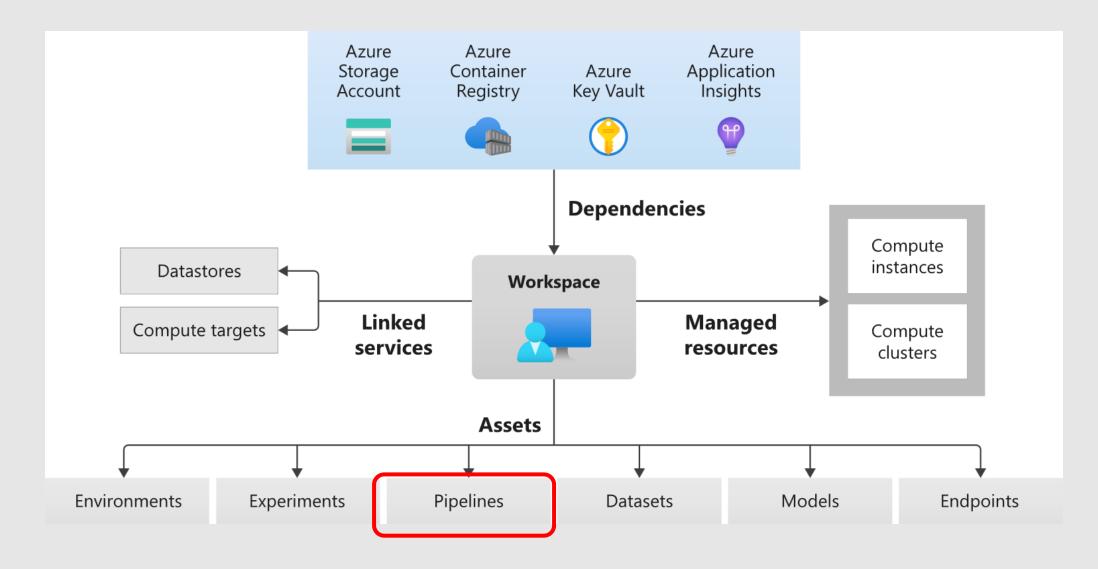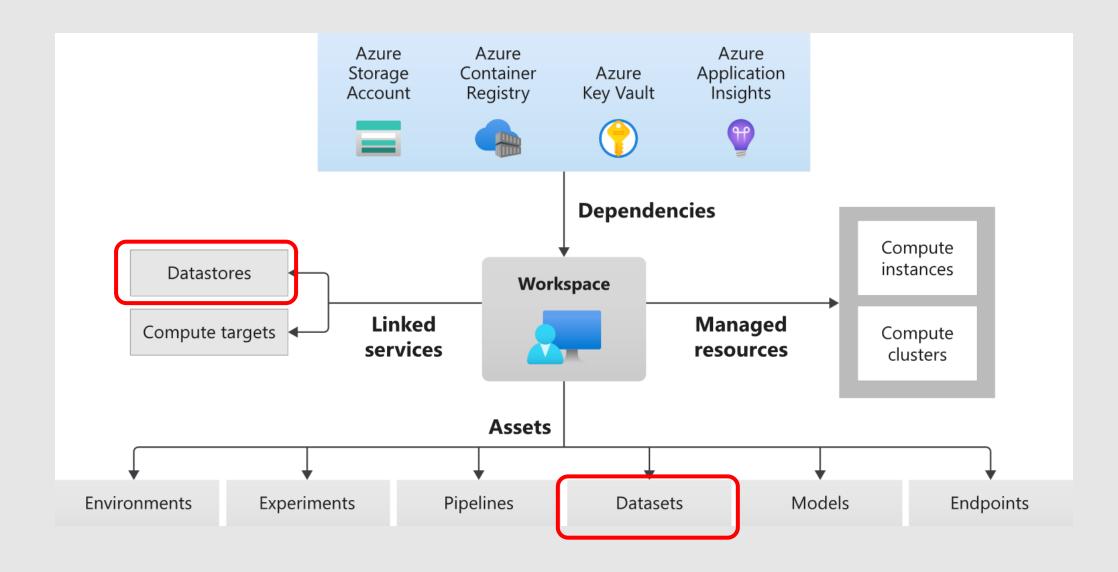
Environments

Experiments

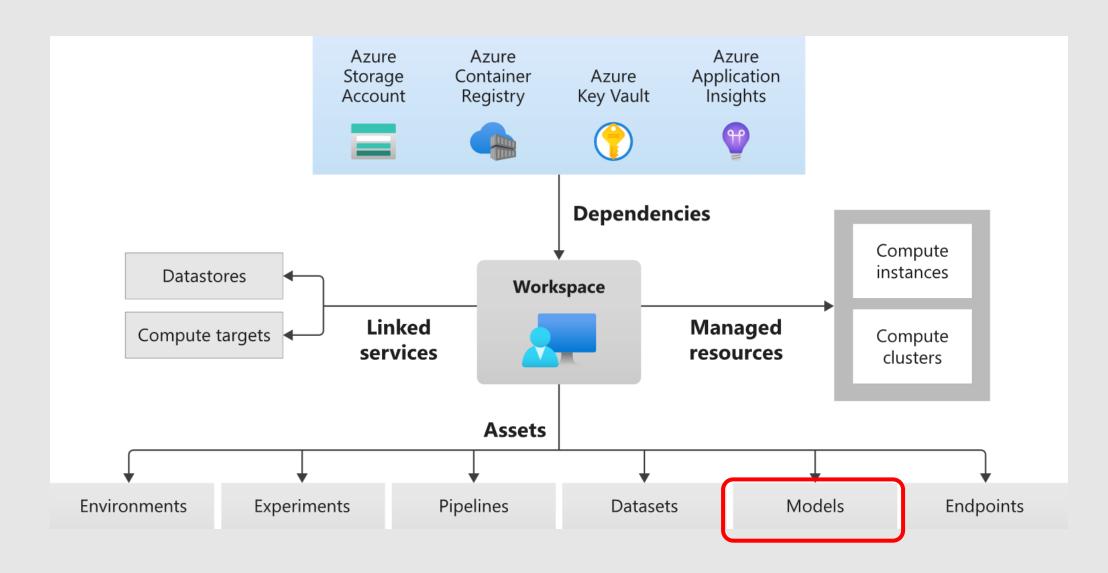# Pipelines
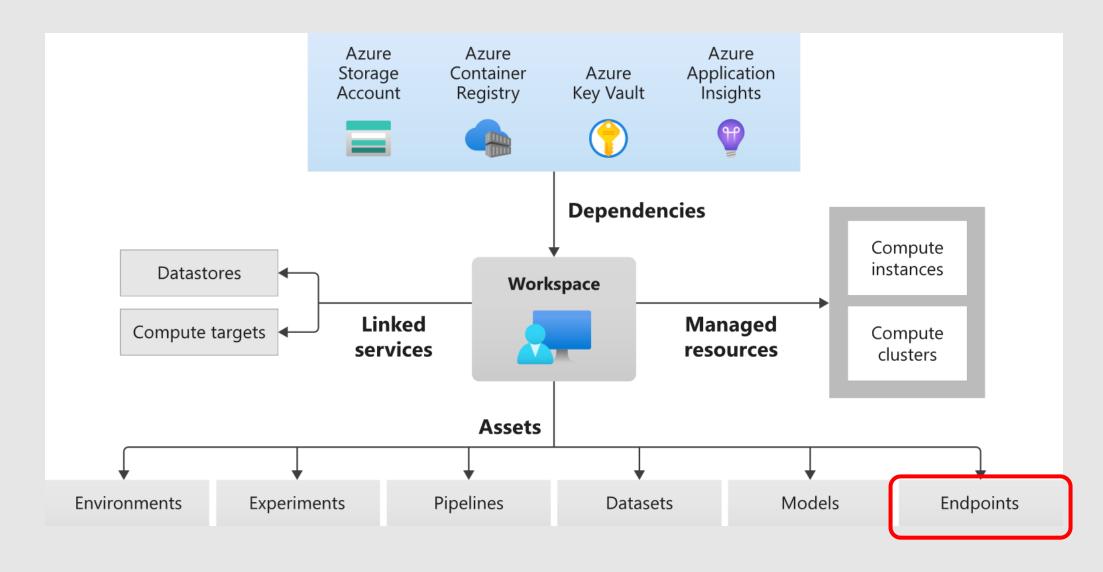
Datastores and Datasets
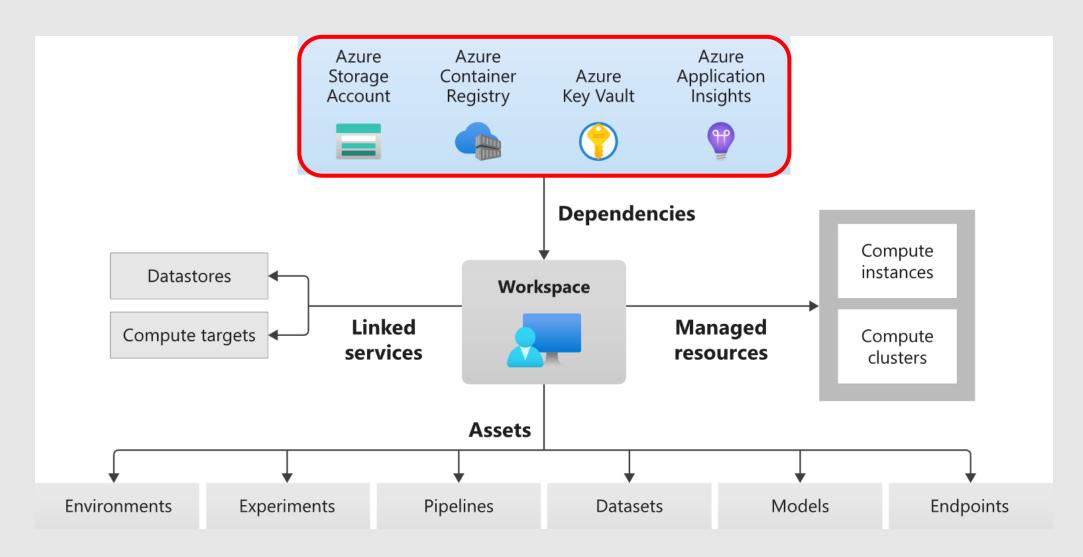
Models

Endpoints

Workspace Services

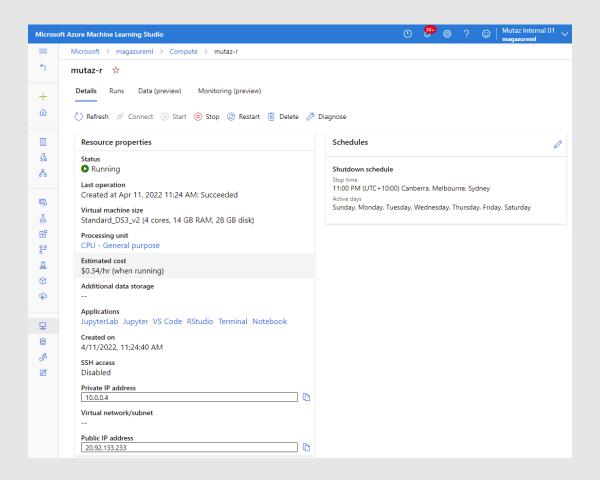# Development Environment Options

Environment	Pros	Cons
Local Environment	- Full control of your development environment and dependencies - Run with any build tool, environment or IDE of your choice	- Takes longer to setup and get started - Require installing SDKs and tools - Compute and storage limits
Remote on Azure ML Compute Instance	- Easy to get started, compute instance preconfigured with tools and libraries - AML SDK notebooks and tutorials preloaded - Scale compute and storage	- Manage cost for compute instance - More complex development environment setup

Compute Instance - CPU

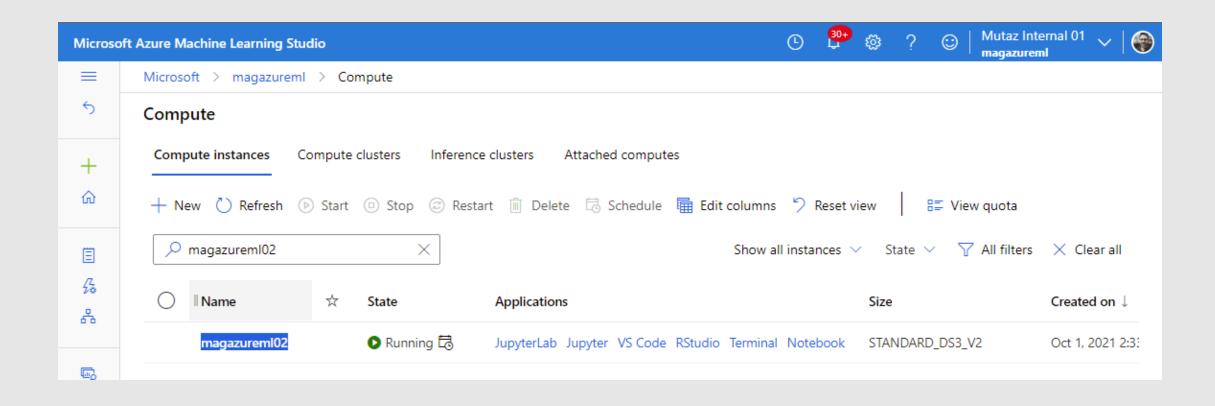- Provision a compute instance:
  - Size: STANDARD_DS3_V2 (4 cores, 14 GB RAM, 28 GB disk)
  - Set shutdown schedule

- Preconfigured with RStudio and Terminal

- Coding Options:
  - Rstudio
  - Notebook

Applications on Compute Instances
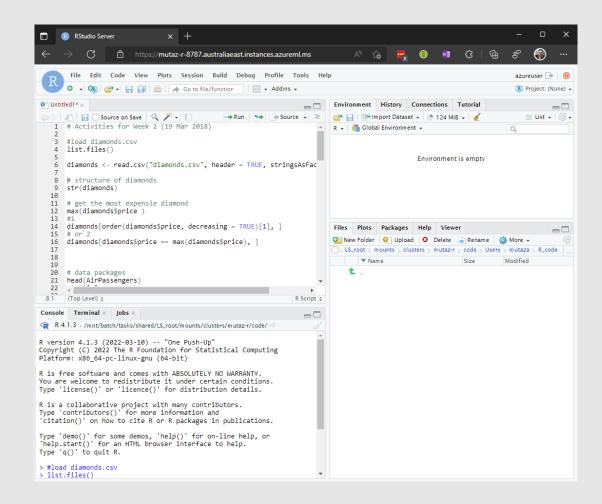
RStudio Online

- You can upload and import a text data files from local computer
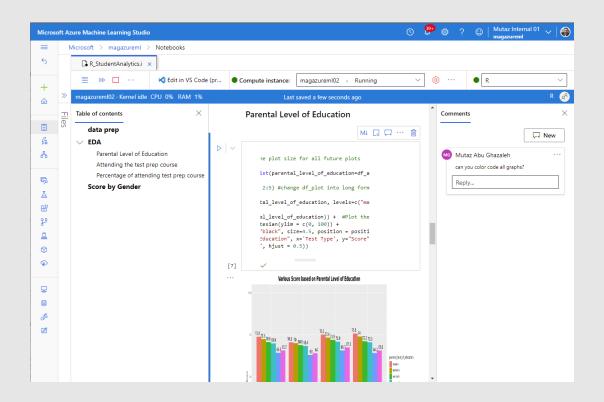
- You can use Git for code management

- Create and run your R scripts

R Notebooks

- Notebook experience for writing and presenting R code

- R code cells are executed against a compute instance, and results are presented inline within the notebook

- Supports markdown, code cells, and results cells

- Co-authoring, notebook comments and table of content
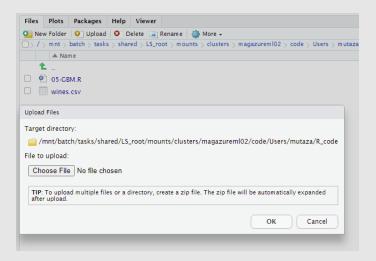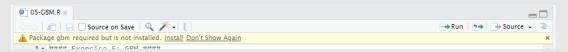
Download Code
https://github.com/mutazag/R_Coding
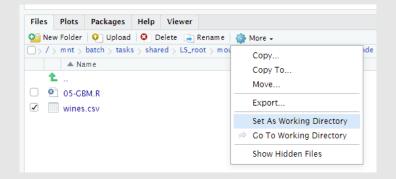
R Code Example

- Open R Studio

- Upload code and data files

- Install required packages (in this example you might need to install gbm)

- Change context to correct code folder:
  - Use the UI tool to "set as working Directory", or
  - Use getwd() and setwd() functions

- Run code

- Solution file: 01-GBM.R
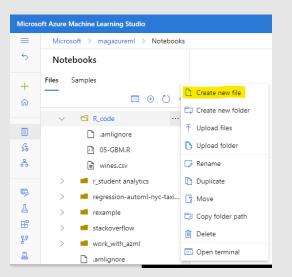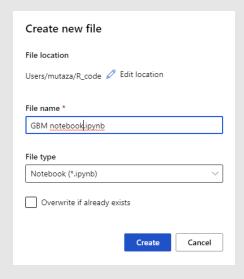
Exercise: R Notebook
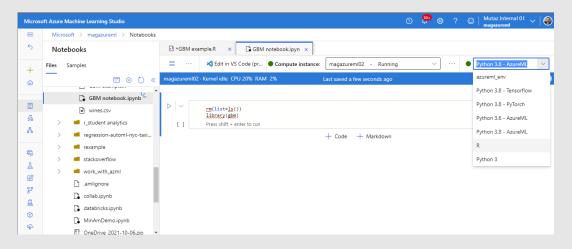
- Implement the same code example as an R Notebook

- Explore with using markdown cells, headings, comments and co-authoring with others

- Solution file: 02-GBM notebook.ipynb

MLFlow

MLflow is an open-source platform for managing the end-to-end machine learning lifecycle.

Mlflow tackles key functions of ML workloads. MLFlow can be used to track experiments to record and compare parameters and results.

MLflow is library-agnostic. You can use it with any machine learning library, and in any programming language, since all functions are accessible through a REST API and CLI. For convenience, the project also includes a Python API, R API, and Java API.

Experiments and Tracking

Logging to Experiments using mlflow

- Install mlflow library for R
  - `install.packages('mlflow')`

- Set mlflow tracking URI
  - URI address of azureml mlflow tracking server
  - Must login to Azure

- Log using mlflow APIs
  - mlflow_log_metric
  - mlflow_log_param
  - mlflow_set_tag

Set mlflow tracking URI

```r
library(mlflow)
system('az login --identity')

new_tracking_uri = sub('azureml://', 'https://', mlflow_get_tracking_uri())

BEARER_TOKEN <- system(
        'az account get-access-token --query accessToken -o tsv',
        intern = TRUE)

Sys.setenv('MLFLOW_TRACKING_URI'=new_tracking_uri)
Sys.setenv('MLFLOW_TRACKING_TOKEN'= BEARER_TOKEN)

print(Sys.getenv('MLFLOW_TRACKING_URI'))
print(substr(Sys.getenv('MLFLOW_TRACKING_TOKEN'), 1, 30))
```

Log using mlflow API

API	Description	Example
mlflow_log_metric	Logs a metric for a run. Metrics are key-value pair that records a single float measure. A metric can be logged several times. The MLflow Backend keeps track of historical metric values.	mlflow_log_metric('R2',.96) mlflow_log_metric('RMSE', 7692)
log_mlflow_param	Logs a parameter for a run. Examples are params and hyperparams used for ML training. A param is a STRING key-value pair. For a run, a single parameter is allowed to be logged only once.	mlflow_log_param('gbm_cv_folds', 5) mlflow_log_param('num_trees', 200)
Mlflow_set_tag	Sets a tag on a run. Tags are run metadata that can be updated during a run and after a run completes.	mlflow_set_tag('
mlflow_end_run	Terminates a run. Attempts to end the current active run if run_id is not specified.	mlflow_end_run()

Exercise: Add logging to experiment code

1. Prepare mlflow by setting up tracking URI
2. Log dataset sizes as metrics
3. Log training job params:
   - use mlflow_log_params
   - Optionally use mlflow_log_batch
4. Log experiment end time as metric
5. Log experiment performance metrics
   - Tip: get metrics using the function confusionMatrix()
6. Don't forget to end experiment run
- Solution file: 03-GBM notebook-mlflow.ipynb

icy_board_8kskxfr2 ✏️ ⭐

🔄 Refresh   🖊️ Connect to compute   ▶️ Resubmit   ⊗ Cancel   🗑️ Delete

Details   Metrics   Images   Child runs   Outputs + logs   Snapshot   Explanations (preview)   Fairness (preview)   Monitoring (preview)

Properties ✏️

Status
✅ Completed

Created
Apr 16, 2022 7:55 PM

Started
Apr 16, 2022 7:55 PM

Duration
8m 49.17s

Compute duration
8m 49.17s

Compute target
None

Run ID
65842778-39d3-4b98-89c7-eed8f5e48755

Script name
--

Created by
Mutaz Abu Ghazaleh

Input datasets
None

Tags ✏️

mlflow.source.name : <console>   mlflow.source.type : LOCAL   mlflow.user : azureuser

Params

cores_num : 3   gbm_cv_folds : 5   gbm_depth : 2   gbm_n_min : 20   gbm_shrinkage : 0.01

num_trees : 200

Metrics

Accuracy
0.812

AccuracyLower
0.79

AccuracyNull
0.808

AccuracyPValue
0.378

AccuracyUpper
0.833